CSCE460301 - Fundamentals of Computer Vision
Homework 2

Ahmed Badr

Fall 2024

Part 1: Hybrid Image

Overview of Hybrid Image Process

A hybrid image is created by combining a low-pass filtered version of one image with a high-pass filtered version of another.

- **Low-pass filtering** removes high-frequency details, making the image appear blurred and smooth, ideal for capturing broader structures.
- **High-pass filtering** retains only the high-frequency details, preserving edges and finer textures.

The "cutoff frequency" parameter, adjusted by the standard deviation of the Gaussian filter, controls the amount of frequency content retained for each image. This combination enables the viewer to perceive different images based on viewing distance, where low frequencies dominate from afar, and high frequencies become visible up close.

1. Hybrid Image Results

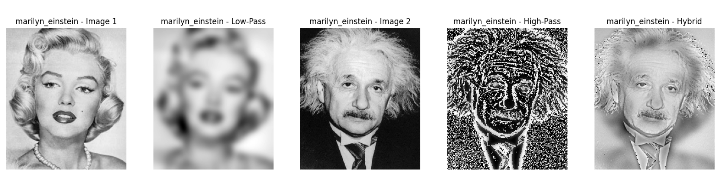

Figure 1: Hybrid Image of Marilyn and Einstein

Marilyn and Einstein Hybrid:

- **Low-pass Filter**: The image of Marilyn is blurred, keeping only the general structure.

- **High-pass Filter**: The image of Einstein retains sharp edges and fine details.
- **Result**: At a distance, Marilyn's features are prominent, while close up, Einstein's detailed facial features are visible.

Figure 2: Hybrid Image of Fish and Submarine

Fish and Submarine Hybrid:

- **Low-pass Filter**: The fish image is blurred to capture the broader structure.
- **High-pass Filter**: The submarine image retains high-frequency details, emphasizing its outline.
- **Result**: From a distance, the fish shape is clear, but the submarine's edges are noticeable up close.

Figure 3: Hybrid Image of Bird and Plane

Bird and Plane Hybrid:

- **Low-pass Filter**: The bird image is blurred, focusing on the silhouette.
- **High-pass Filter**: The plane's details are emphasized through edge retention.
- **Result**: At different distances, the image shifts between resembling a bird and a plane.

2. Fourier Transform Analysis

The Fourier transform highlights the frequency distribution in the images:

- The low-pass filtered images have fewer high frequencies, as shown by the compact Fourier transform pattern.
- The high-pass images contain spread-out high frequencies due to edge emphasis.

These Fourier transforms help illustrate how each filtered image contributes different frequency ranges to the hybrid image.

3. Code Improvements and Adjustments

To further improve the results, several enhancements to the code can be made:

- **Optimize Cutoff Frequency**: Experimenting with different values of the Gaussian filter's standard deviation for low-pass and high-pass filters can help achieve a clearer image.
- **Image Alignment**: Properly aligning the images before creating hybrids improves the coherence of the final output.
- **Color Enhancement** (optional): Adding color selectively to the low-pass or high-pass component can further enhance the hybrid effect. This could be tested by converting each filtered image back to color and blending them.

4. Summary

The hybrid images created illustrate how different frequency ranges can be combined to present unique effects. Experimenting with the cutoff frequencies allows us to fine-tune the appearance of each hybrid image, making certain features more prominent at various distances. Further improvements can be made through alignment, parameter tuning, and color enhancement for visually enriched results.

Part 2: Image Pyramid

Overview of Gaussian and Laplacian Pyramids

An image pyramid is a multiscale representation of an image. The Gaussian pyramid involves progressively applying a Gaussian blur and downsampling the image, reducing resolution at each level. The Laplacian pyramid is created by subtracting the blurred image (Gaussian) from the original, isolating high-frequency details at each level.

1. Gaussian Pyramid

The Gaussian pyramid displays the progressively blurred images at different scales. Here are the results for an image with 5 pyramid levels:

Figure 4: Gaussian Pyramid of Image

2. FFT Amplitudes of Gaussian Pyramid

To analyze the frequency content of the Gaussian pyramid, we compute the Fourier Transform at each level. This shows how frequency changes as the image is blurred:

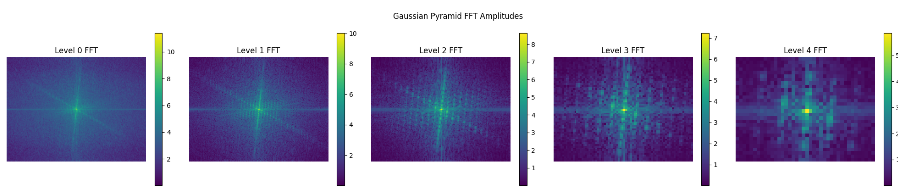

Figure 5: FFT Amplitudes of Gaussian Pyramid

3. Laplacian Pyramid

The Laplacian pyramid isolates high-frequency components at each level by subtracting the blurred (Gaussian) image from the original. Here is the Laplacian pyramid with 5 levels:

Figure 6: Laplacian Pyramid of Image

4. FFT Amplitudes of Laplacian Pyramid

The Fourier Transform for the Laplacian pyramid shows the preserved frequency components at each level:

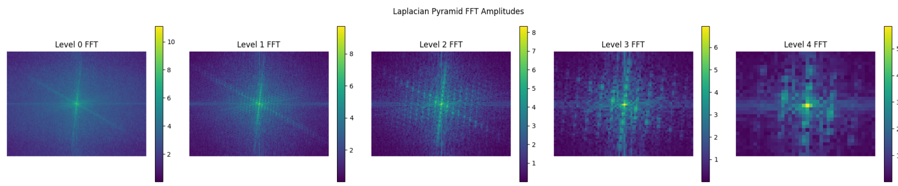

Figure 7: FFT Amplitudes of Laplacian Pyramid

5. Explanation of Gaussian and Laplacian Pyramids

- **Gaussian Pyramid**: At each level, the image becomes progressively more blurred, removing high-frequency details and reducing resolution. This allows the image to be viewed at multiple scales, where lower-resolution levels focus on broader structures and remove finer details.
- **Laplacian Pyramid**: The Laplacian pyramid preserves edges and fine details by subtracting the blurred version of the image from the original. This reveals the high-frequency details that the Gaussian filter removes.
- **FFT Analysis**: The FFT amplitudes reveal the frequency content at each level. For the Gaussian pyramid, lower frequencies dominate at higher pyramid levels. In the Laplacian pyramid, the frequencies preserved become more localized at each level, focusing on high-frequency details.

Part 3: Edge Detection

1. Gradient-Based Edge Detection

We built a simple gradient-based edge detector that computes the gradient magnitude and orientation for each pixel in an image. This edge detector uses Gaussian smoothing, Sobel filters for gradients, and non-maximal suppression to generate a final soft edge map.

Function: gradientMagnitude(im, sigma)

- This function takes an image as input, smooths it using a Gaussian filter, computes gradients in the x and y directions, and returns the magnitude and orientation of the gradient.
- The gradients are computed using Sobel filters, which are applied to the smoothed image.

Function: edgeGradient(im)

- This function calls the gradientMagnitude function, then applies non-maximal suppression using the magnitudes along the binary edges produced by the Canny edge detector to generate a soft boundary map.

2. Oriented Filter-Based Edge Detection

We extended the edge detection process using oriented filters, which can capture edges in multiple orientations and improve the accuracy of the edge maps.

Function: orientedFilterMagnitude(im, orientations=4)

- This function computes edge magnitudes and orientations using a set of Gabor filters, each oriented at a different angle (e.g., 0°, 45°, 90°, 135°).
- The maximum response from all filters at each pixel is taken to generate the final edge map, which preserves edges at different angles.

Function: edgeOrientedFilters(im, orientations=4)

- This function calls orientedFilterMagnitude and applies non-maximal suppression to generate the final soft edge map.

3. Results

Example 1: Simple Gradient-Based Edge Detection

Figure 8: Gradient-Based Edge Detection Result

Example 2: Oriented Filter-Based Edge Detection

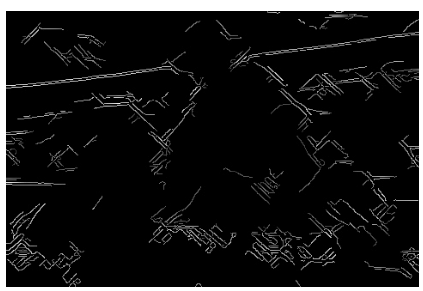

Figure 9: Oriented Filter-Based Edge Detection Result

4. Precision-Recall Plots and F-Scores

The performance of the edge detectors was evaluated using the BSDS dataset and the evaluation script provided. Below are the precision-recall plots for both the simple gradient-based edge detector and the oriented filter-based edge detector.

Precision-Recall Plot for Gradient-Based Edge Detector

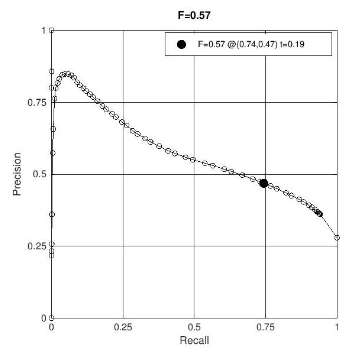

Figure 10: Precision-Recall Plot for Gradient-Based Edge Detection

Precision-Recall Plot for Oriented Filter-Based Edge Detector

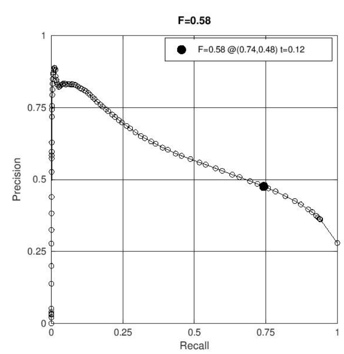

Figure 11: Precision-Recall Plot for Oriented Filter-Based Edge Detection

5. Quantitative Results (F-Scores)

The overall and average F-scores were calculated for each method, summarizing the detector's performance across the test images. The F-scores reflect the trade-off between precision and recall for each detector. The results are shown below.

Method	Overall F-Score	Average F-Score
Gradient-Based	0.57	0.62
Oriented Filters	0.58	0.63

Table 1: F-Scores for Edge Detection Methods

6. Design Choices and Parameters

- **Gaussian Smoothing**: We used a standard deviation ('sigma') of 1.0 for the Gaussian smoothing in both the gradient-based and oriented filter detectors.
- **Oriented Filters**: For the oriented filter-based method, we used four orientations (0°, 45°, 90°, 135°) with Gabor filters, which are designed to capture edges in these directions.

7. Explanation of Results

- **Gradient-Based Detector**: The simple gradient-based detector works well for detecting sharp transitions in intensity, but it is limited in its ability to capture edges at multiple orientations.
- **Oriented Filter Detector**: The oriented filter-based detector improves performance slightly by capturing edges at multiple orientations, resulting in a higher F-score.
- **Precision-Recall Analysis**: The precision-recall curves show that both detectors perform similarly, but the oriented filters improve edge detection in images with complex textures and varying orientations.